NAG Library Routine Document

F08HQF (ZHBEVD)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

Warning. The specification of the arguments LRWORK and LIWORK changed at Mark 20 in the case where JOB = 'V' and N > 1: the minimum dimension of the array RWORK has been reduced whereas the minimum dimension of the array IWORK has been increased.

1 Purpose

F08HQF (ZHBEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian band matrix. If the eigenvectors are requested, then it uses a divide-and-conquer algorithm to compute eigenvalues and eigenvectors. However, if only eigenvalues are required, then it uses the Pal–Walker–Kahan variant of the QL or QR algorithm.

2 Specification

```
SUBROUTINE F08HQF (JOB, UPLO, N, KD, AB, LDAB, W, Z, LDZ, WORK, LWORK,     &
                   RWORK, LRWORK, IWORK, LIWORK, INFO)
INTEGER                 N, KD, LDAB, LDZ, LWORK, LRWORK,                   &
                        IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp)      W(*), RWORK(max(1,LRWORK))
COMPLEX (KIND=nag_wp)   AB(LDAB,*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)            JOB, UPLO
```

The routine may be called by its LAPACK name ***zhbevd***.

3 Description

F08HQF (ZHBEVD) computes all the eigenvalues and, optionally, all the eigenvectors of a complex Hermitian band matrix A. In other words, it can compute the spectral factorization of A as

$$A = Z\Lambda Z^{\mathrm{H}},$$

where Λ is a real diagonal matrix whose diagonal elements are the eigenvalues λ_i, and Z is the (complex) unitary matrix whose columns are the eigenvectors z_i. Thus

$$Az_i = \lambda_i z_i, \quad i = 1, 2, \ldots, n.$$

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Arguments

1: JOB – CHARACTER(1) *Input*

On entry: indicates whether eigenvectors are computed.

JOB = 'N'
Only eigenvalues are computed.

JOB = 'V'
Eigenvalues and eigenvectors are computed.

Constraint: JOB = 'N' or 'V'.

2: UPLO – CHARACTER(1) *Input*

On entry: indicates whether the upper or lower triangular part of A is stored.

UPLO = 'U'
The upper triangular part of A is stored.

UPLO = 'L'
The lower triangular part of A is stored.

Constraint: UPLO = 'U' or 'L'.

3: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

4: KD – INTEGER *Input*

On entry: if UPLO = 'U', the number of superdiagonals, k_d, of the matrix A.

If UPLO = 'L', the number of subdiagonals, k_d, of the matrix A.

Constraint: $\mathrm{KD} \geq 0$.

5: AB(LDAB, $*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array AB must be at least $\max(1, \mathrm{N})$.

On entry: the upper or lower triangle of the n by n Hermitian band matrix A.

The matrix is stored in rows 1 to $k_d + 1$, more precisely,

if UPLO = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(k_d + 1 + i - j, j)$ for $\max(1, j - k_d) \leq i \leq j$;

if UPLO = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_d)$.

On exit: AB is overwritten by values generated during the reduction to tridiagonal form.

The first superdiagonal or subdiagonal and the diagonal of the tridiagonal matrix T are returned in AB using the same storage format as described above.

6: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F08HQF (ZHBEVD) is called.

Constraint: $\mathrm{LDAB} \geq \mathrm{KD} + 1$.

7: W($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array W must be at least $\max(1, \mathrm{N})$.

On exit: the eigenvalues of the matrix A in ascending order.

8: Z(LDZ, $*$) – COMPLEX (KIND=nag_wp) array *Output*

Note: the second dimension of the array Z must be at least $\max(1, N)$ if JOB = 'V' and at least 1 if JOB = 'N'.

On exit: if JOB = 'V', Z is overwritten by the unitary matrix Z which contains the eigenvectors of A. The ith column of Z contains the eigenvector which corresponds to the eigenvalue $W(i)$.

If JOB = 'N', Z is not referenced.

9: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08HQF (ZHBEVD) is called.

Constraints:

if JOB = 'V', $LDZ \geq \max(1, N)$;
if JOB = 'N', $LDZ \geq 1$.

10: WORK($\max(1, LWORK)$) – COMPLEX (KIND=nag_wp) array *Workspace*

On exit: if $INFO = 0$, the real part of WORK(1) contains the required minimal size of LWORK.

11: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08HQF (ZHBEVD) is called.

If $LWORK = -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraints:

if $N \leq 1$, $LWORK \geq 1$ or $LWORK = -1$;
if JOB = 'N' and $N > 1$, $LWORK \geq N$ or $LWORK = -1$;
if JOB = 'V' and $N > 1$, $LWORK \geq 2 \times N^2$ or $LWORK = -1$.

12: RWORK($\max(1, LRWORK)$) – REAL (KIND=nag_wp) array *Workspace*

On exit: if $INFO = 0$, RWORK(1) contains the required minimal size of LRWORK.

13: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which F08HQF (ZHBEVD) is called.

If $LRWORK = -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the RWORK array, returns this value as the first entry of the RWORK array, and no error message related to LRWORK is issued.

Constraints:

if $N \leq 1$, $LRWORK \geq 1$ or $LRWORK = -1$;
if JOB = 'N' and $N > 1$, $LRWORK \geq N$ or $LRWORK = -1$;
if JOB = 'V' and $N > 1$, $LRWORK \geq 2 \times N^2 + 5 \times N + 1$ or $LRWORK = -1$.

14: IWORK($\max(1, LIWORK)$) – INTEGER array *Workspace*

On exit: if $INFO = 0$, IWORK(1) contains the required minimal size of LIWORK.

15: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08HQF (ZHBEVD) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the IWORK array, returns this value as the first entry of the IWORK array, and no error message related to LIWORK is issued.

Constraints:

if JOB = 'N' or $\mathrm{N} \leq 1$, $\mathrm{LIWORK} \geq 1$ or $\mathrm{LIWORK} = -1$;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LIWORK} \geq 5 \times \mathrm{N} + 3$ or $\mathrm{LIWORK} = -1$.

16: INFO – INTEGER *Output*

On exit: INFO $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

if INFO $= i$ and JOB = 'N', the algorithm failed to converge; i elements of an intermediate tridiagonal form did not converge to zero; if INFO $= i$ and JOB = 'V', then the algorithm failed to compute an eigenvalue while working on the submatrix lying in rows and column $i/(\mathrm{N}+1)$ through $i \bmod (\mathrm{N}+1)$.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $(A+E)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***. See Section 4.7 of Anderson *et al.* (1999) for further details.

8 Parallelism and Performance

F08HQF (ZHBEVD) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08HQF (ZHBEVD) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The real analogue of this routine is F08HCF (DSBEVD).

10 Example

This example computes all the eigenvalues and eigenvectors of the Hermitian band matrix A, where

$$A = \begin{pmatrix} 1+0i & 2-1i & 3-1i & 0+0i & 0+0i \\ 2+1i & 2+0i & 3-2i & 4-2i & 0+0i \\ 3+1i & 3+2i & 3+0i & 4-3i & 5-3i \\ 0+0i & 4+2i & 4+3i & 4+0i & 5-4i \\ 0+0i & 0+0i & 5+3i & 5+4i & 5+0i \end{pmatrix}.$$

10.1 Program Text

```
    Program f08hqfe

!     F08HQF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dznrm2, nag_wp, x04daf, zhbevd
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)           :: scal
      Integer                         :: i, ifail, info, j, k, kd, ldab, ldz, &
                                         liwork, lrwork, lwork, n
      Character (1)                   :: job, uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ab(:,:), work(:), z(:,:)
      Real (Kind=nag_wp), Allocatable  :: rwork(:), w(:)
      Integer, Allocatable            :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                       :: abs, conjg, max, maxloc, min
!     .. Executable Statements ..
      Write (nout,*) 'F08HQF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, kd
      ldab = n
      ldz = n
      liwork = 5*n + 3
      lrwork = 2*n*n + 5*n + 1
      lwork = 2*n*n
      Allocate (ab(ldab,n),work(lwork),z(ldz,n),rwork(lrwork),w(n),            &
        iwork(liwork))

!     Read A from data file

      Read (nin,*) uplo
      If (uplo=='U') Then
        Do i = 1, n
          Read (nin,*)(ab(kd+1+i-j,j),j=i,min(n,i+kd))
        End Do
      Else If (uplo=='L') Then
        Do i = 1, n
          Read (nin,*)(ab(1+i-j,j),j=max(1,i-kd),i)
        End Do
      End If

      Read (nin,*) job

!     Calculate all the eigenvalues and eigenvectors of A
!     The NAG name equivalent of zhbevd is f08hqf
      Call zhbevd(job,uplo,n,kd,ab,ldab,w,z,ldz,work,lwork,rwork,lrwork,iwork, &
        liwork,info)

      Write (nout,*)
      If (info>0) Then
        Write (nout,*) 'Failure to converge.'
      Else

!       Print eigenvalues and eigenvectors

        Write (nout,*) 'Eigenvalues'
        Do i = 1, n
          Write (nout,99999) i, w(i)
        End Do
        Write (nout,*)
        Flush (nout)
```

```
!       Normalize the eigenvectors, largest element real
        Do i = 1, n
          rwork(1:n) = abs(z(1:n,i))
          k = maxloc(rwork(1:n),1)
          scal = conjg(z(k,i))/abs(z(k,i))/dznrm2(n,z(1,i),1)
          z(1:n,i) = z(1:n,i)*scal
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04daf('General',' ',n,n,z,ldz,'Eigenvectors',ifail)

      End If

99999 Format (3X,I5,5X,2F8.4)
    End Program f08hqfe
```

10.2 Program Data

```
F08HQF Example Program Data
  5  2                                                   :Values of N and KD
  'L'                                                    :Value of UPLO
  (1.0, 0.0)
  (2.0, 1.0) (2.0, 0.0)
  (3.0, 1.0) (3.0, 2.0) (3.0, 0.0)
             (4.0, 2.0) (4.0, 3.0) (4.0, 0.0)
                        (5.0, 3.0) (5.0, 4.0) (5.0, 0.0) :End of matrix A
  'V'                                                    :Value of JOB
```

10.3 Program Results

```
 F08HQF Example Program Results

 Eigenvalues
       1      -6.4185
       2      -1.4094
       3       1.4421
       4       4.4856
       5      16.9002

 Eigenvectors
          1       2       3       4       5
 1  -0.2534  0.6367 -0.2560  0.0171  0.1051
    -0.0538  0.0000  0.3721  0.5500 -0.0983

 2  -0.0662 -0.2578  0.5344 -0.2608  0.2516
     0.4301  0.2413  0.0000  0.4869 -0.1789

 3   0.5274 -0.3039 -0.4245 -0.0399  0.4994
     0.0000 -0.3481  0.0915  0.2142 -0.1513

 4   0.1061  0.3450  0.4964 -0.0253  0.5611
    -0.4981 -0.0832 -0.1546 -0.1700  0.0000

 5  -0.4519 -0.2469 -0.1979  0.5614  0.4837
     0.0424  0.2634 -0.1114  0.0000  0.2509
```